(12) UK Patent Application (19) GB (11) 2 176 918 A

(43) Application published 7 Jan 1987

(21) Application No 8519991

(22) Date of filing 8 Aug 1985

(30) Priority data

(31) 744389 (32) 13 June 1985 (33) US

(51) INT CL[4]
G06F 12/10 // 12/14

(52) Domestic classification (Edition I):
G4A MA NV

(56) Documents cited
GB A 2127994 GB 1595740

(58) Field of search
G4A
Selected US specifications from IPC sub-class G06F

(71) Applicant
Intel Corporation,

(Incorporated in USA-California),

3065 Bowers Avenue, Santa Clara, California 95051, United States of America

(72) Inventors
John H. Crawford,
Paul S. Ries

(74) Agent and/or Address for Service
Potts, Kerr & Co., 15 Hamilton Square, Birkenhead, Merseyside L41 6BR

(54) Memory management for microprocessor system

(57) Microprocessor architecture for an address translation unit provides two levels of cache memory. Segmentation registers 21 and an associated segmentation table in main memory provide a first level of memory management which includes attribute bits used for protection, priority, etc. A second page level cache memory 22 which includes an associated page directory and page table in main memory provide a second level of management with independent protection on a page level.

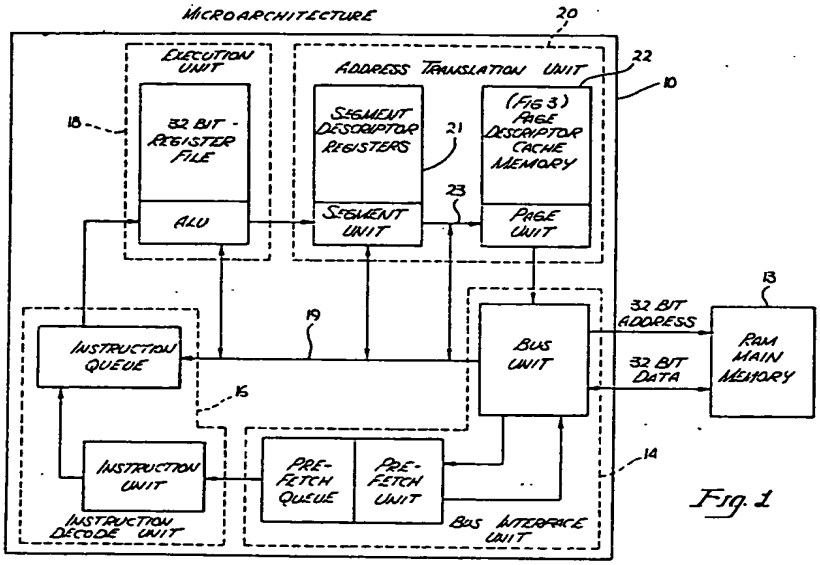

Fig. 1

SEE ERRATA SLIP ATTACHED

SEE ERRATA SLIP ATTACHED

GB 2 176 918 A

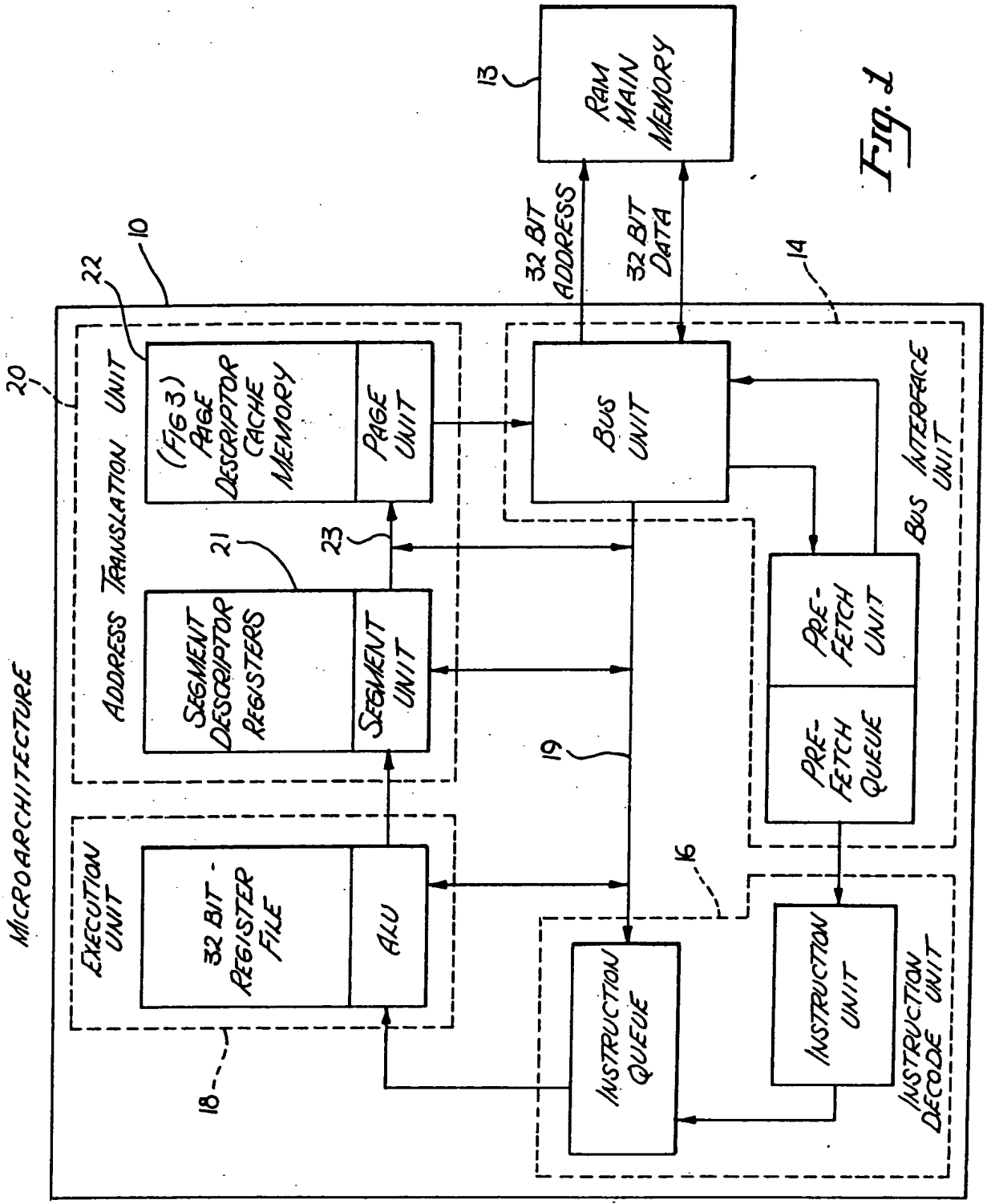
MICROARCHITECTURE
20
10
22
ADDRESS TRANSLATION UNIT
(FIG 3) PAGE DESCRIPTOR CACHE MEMORY
PAGE UNIT
21
SEGMENT DESCRIPTOR REGISTERS
SEGMENT UNIT
23
EXECUTION UNIT
32 BIT REGISTER FILE
ALU
18
19
16
INSTRUCTION QUEUE
INSTRUCTION UNIT
INSTRUCTION DECODE UNIT
BUS UNIT
BUS INTERFACE UNIT
PRE-FETCH UNIT
PRE-FETCH QUEUE
14
32 BIT ADDRESS
32 BIT DATA
13
RAM MAIN MEMORY

Fig. 1

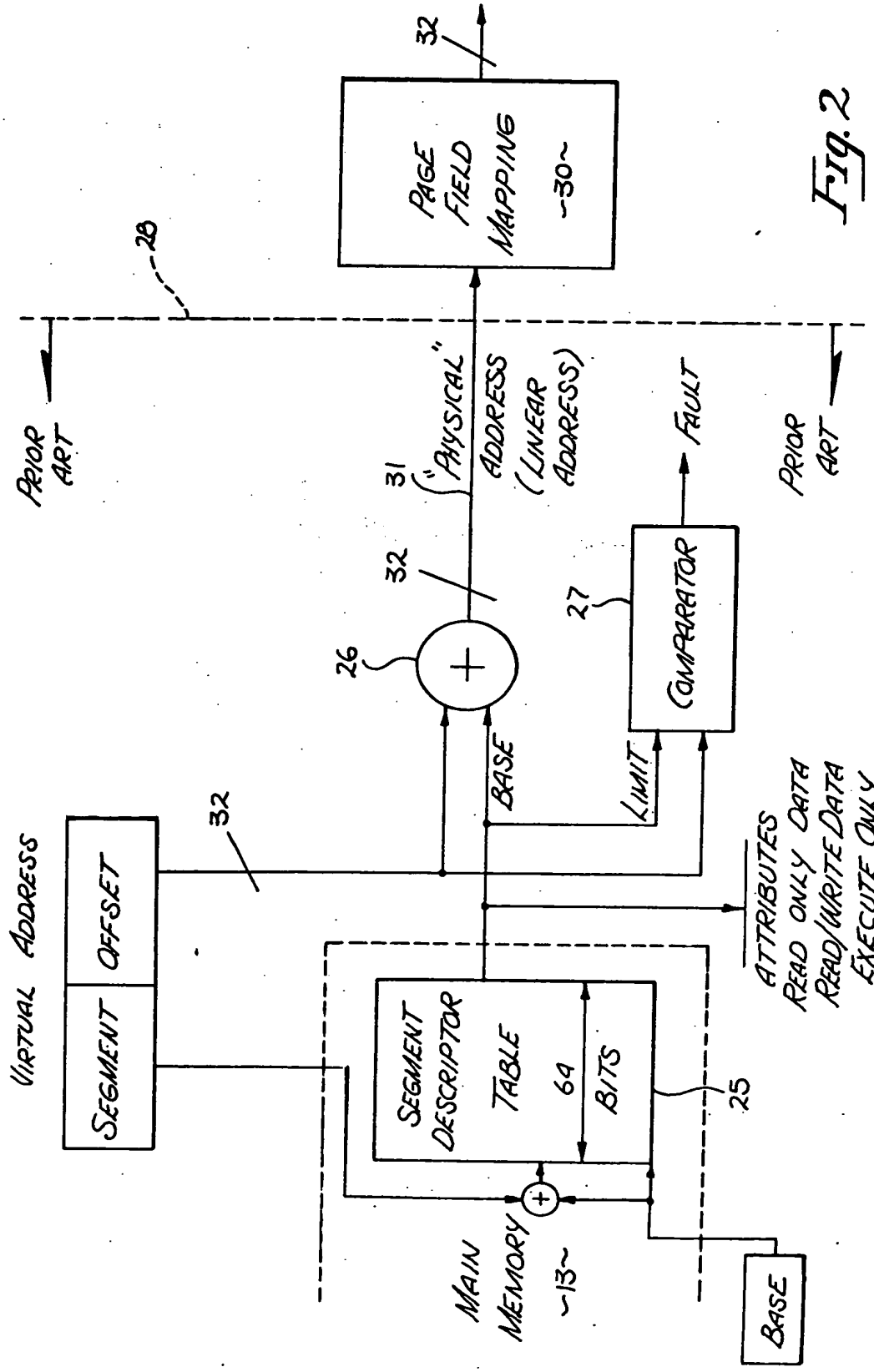
32
PAGE FIELD MAPPING
~30~
28
PRIOR ART
31
"PHYSICAL" ADDRESS (LINEAR ADDRESS)
FAULT
PRIOR ART
32
26
27
COMPARATOR
BASE
LIMIT
VIRTUAL ADDRESS
OFFSET
SEGMENT
32
ATTRIBUTES
READ ONLY DATA
READ/WRITE DATA
EXECUTE ONLY
SEGMENT DESCRIPTOR TABLE
64 BITS
25
MAIN MEMORY
~13~
BASE

Fig. 2

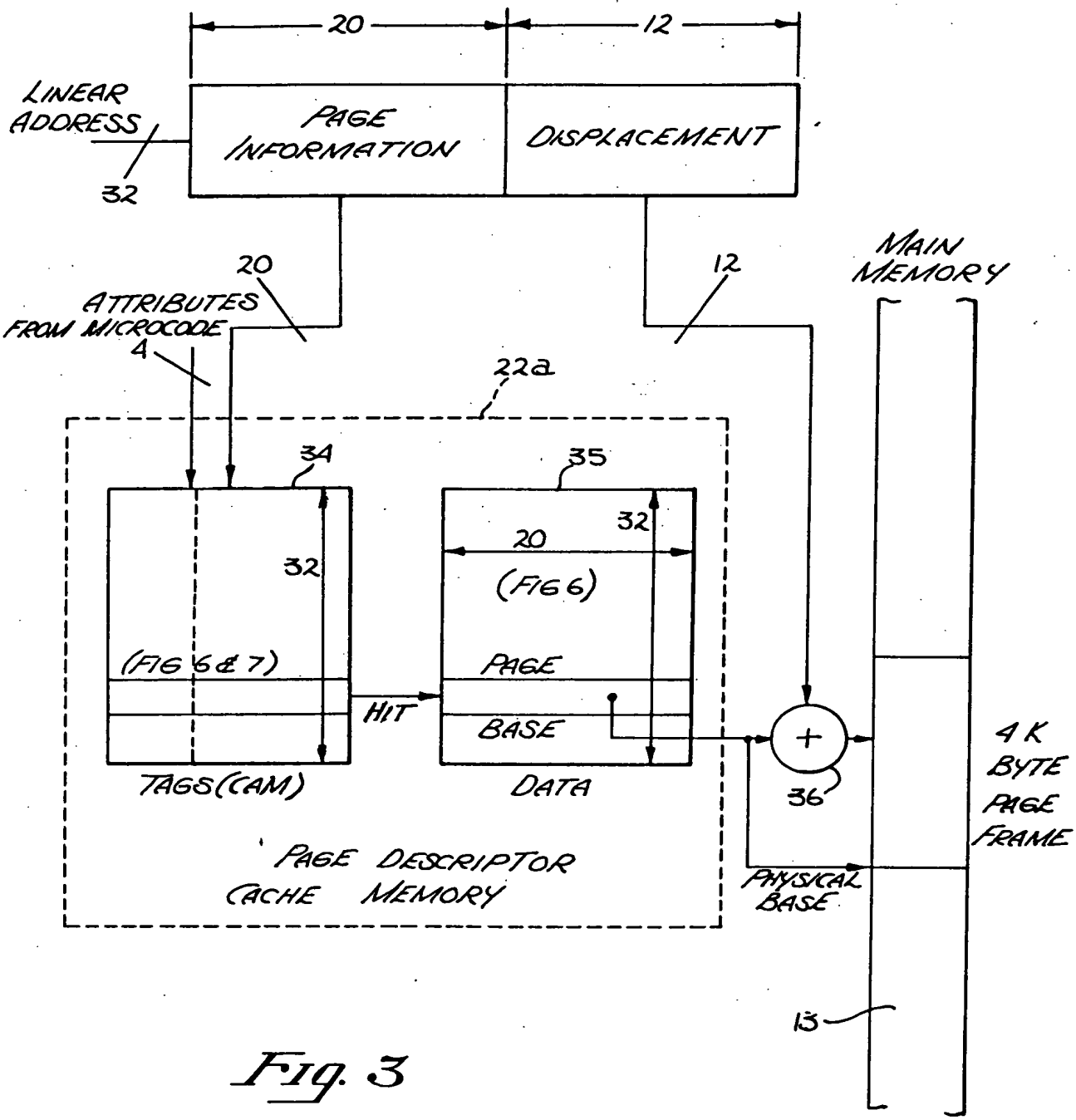
20
12
LINEAR ADDRESS
32
PAGE INFORMATION
DISPLACEMENT
20
ATTRIBUTES FROM MICROCODE
4
12
MAIN MEMORY
22a
34
35
32
20
(FIG 6)
32
(FIG 6 & 7)
PAGE
HIT
BASE
TAGS (CAM)
DATA
36
4K BYTE PAGE FRAME
PAGE DESCRIPTOR CACHE MEMORY
PHYSICAL BASE
13

Fig. 3

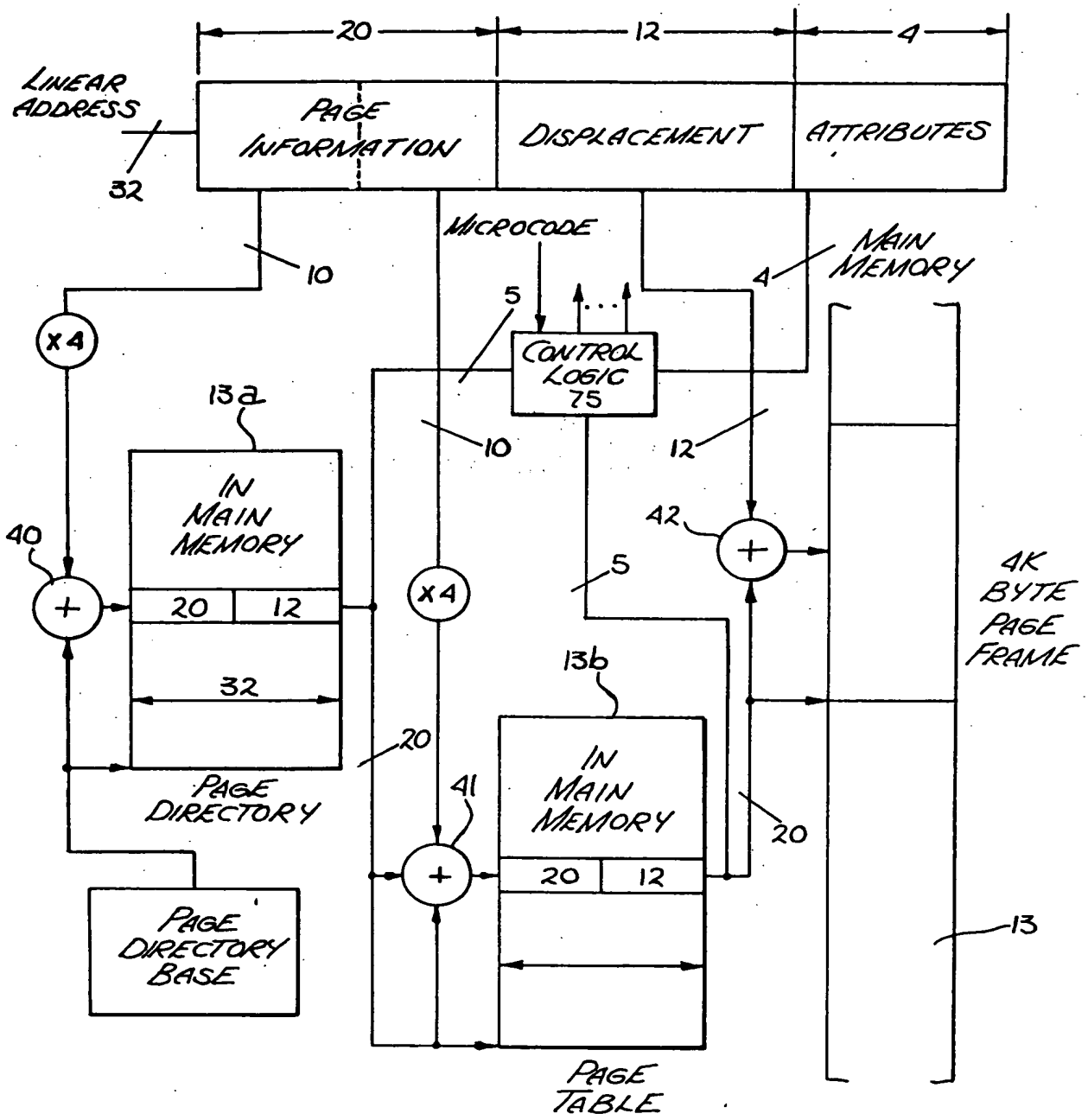
20
12
4
LINEAR ADDRESS
32
PAGE INFORMATION
DISPLACEMENT
ATTRIBUTES
MICROCODE
MAIN MEMORY
10
4
5
X4
CONTROL LOGIC 75
13a
10
12
IN MAIN MEMORY
40
42
20
12
X4
5
4K BYTE PAGE FRAME
32
13b
20
IN MAIN MEMORY
41
20
PAGE DIRECTORY
PAGE DIRECTORY BASE
20
12
13
PAGE TABLE

Fig. 4

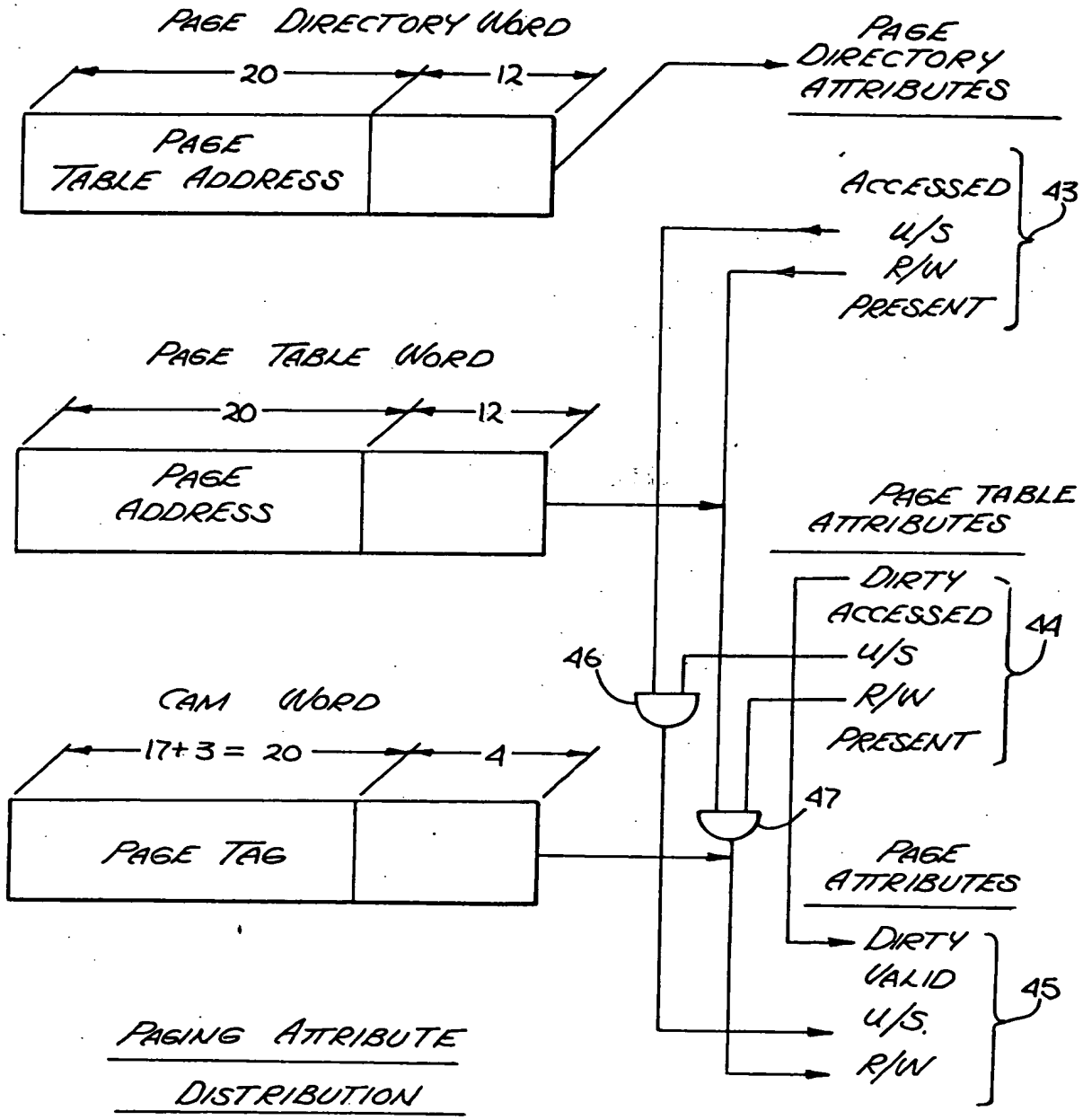
PAGE DIRECTORY WORD
20
12
PAGE TABLE ADDRESS
PAGE DIRECTORY ATTRIBUTES
ACCESSED
U/S
R/W
PRESENT
43
PAGE TABLE WORD
20
12
PAGE ADDRESS
PAGE TABLE ATTRIBUTES
DIRTY
ACCESSED
U/S
R/W
PRESENT
44
46
CAM WORD
17+3 = 20
4
PAGE TAG
47
PAGE ATTRIBUTES
DIRTY
VALID
U/S.
R/W
45
PAGING ATTRIBUTE DISTRIBUTION

Fig. 5.

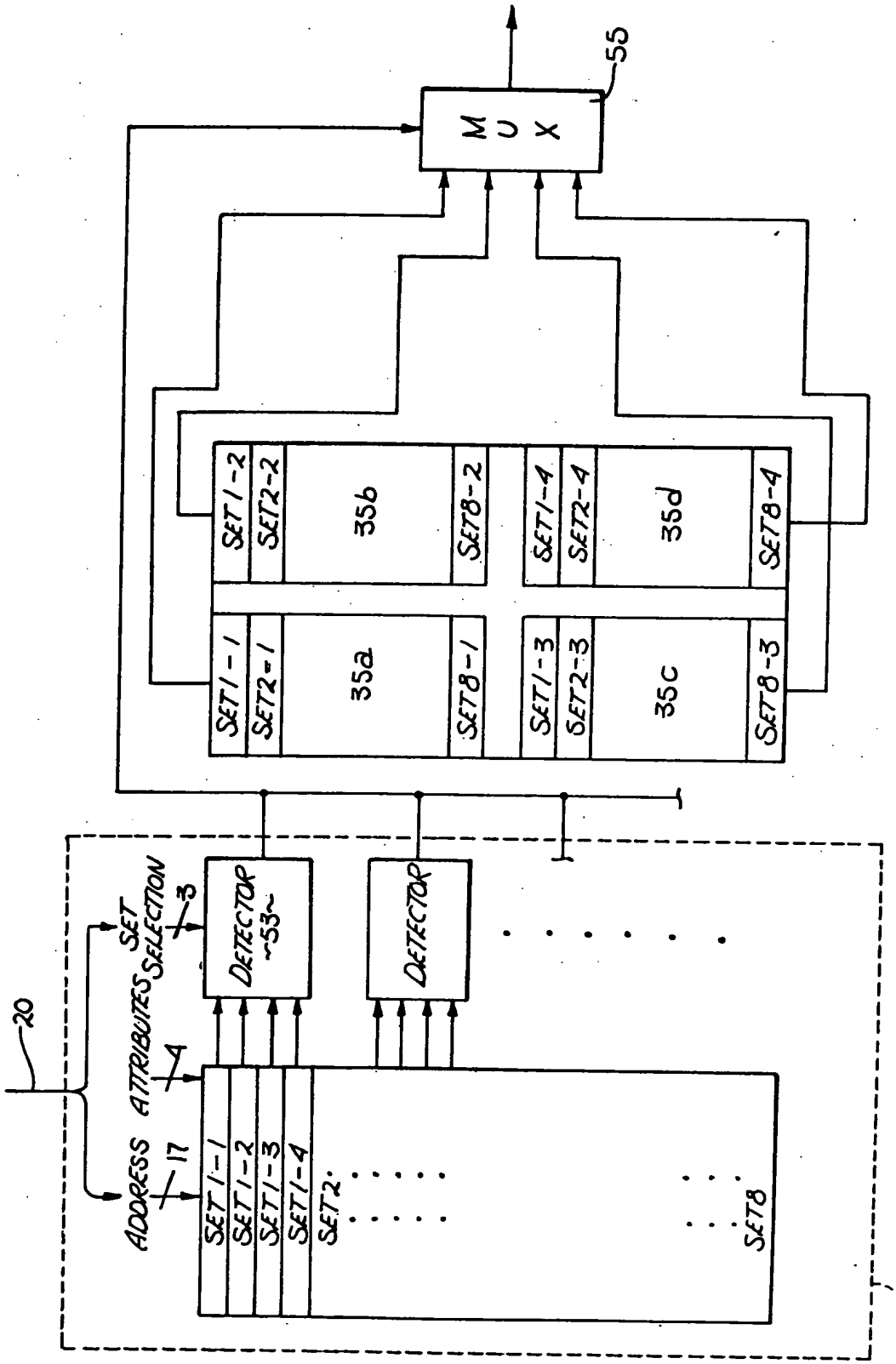
MUX
55
SET 1-2
SET 2-2
35b
SET 8-2
SET 1-4
SET 2-4
35d
SET 8-4
SET 1-1
SET 2-1
35a
SET 8-1
SET 1-3
SET 2-3
35c
SET 8-3
SET SELECTION
3
DETECTOR
53
DETECTOR
20
ATTRIBUTES
4
ADDRESS
17
SET 1-1
SET 1-2
SET 1-3
SET 1-4
SET 2
SET 8
34

Fig. 6

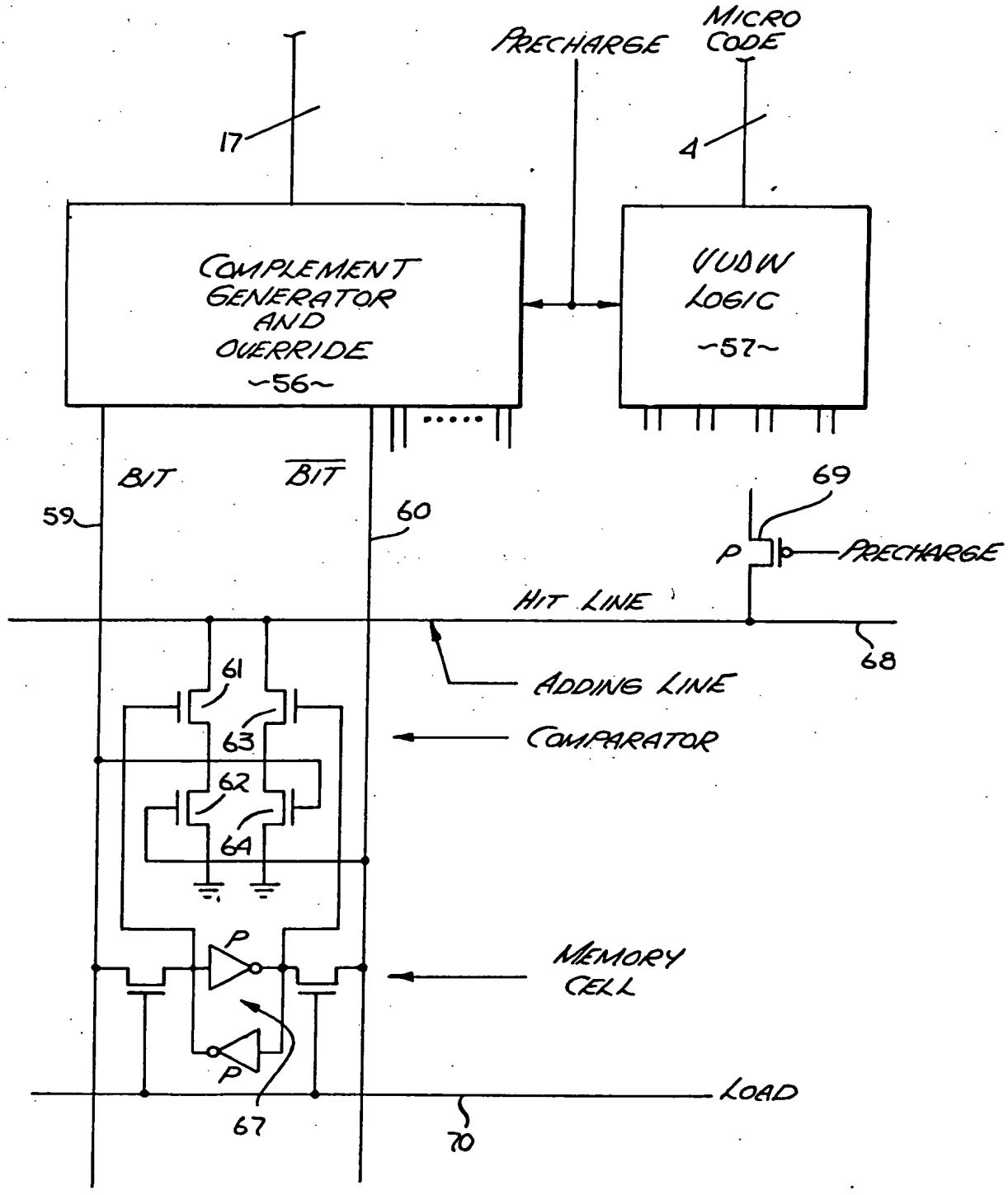
PRECHARGE
MICRO CODE
17
4
COMPLEMENT GENERATOR AND OVERRIDE
~56~
VUSW LOGIC
~57~
BIT
BIT
59
60
69
P
PRECHARGE
HIT LINE
68
61
63
62
6A
ADDING LINE
COMPARATOR
P
P
MEMORY CELL
LOAD
67
70

Fig. 7

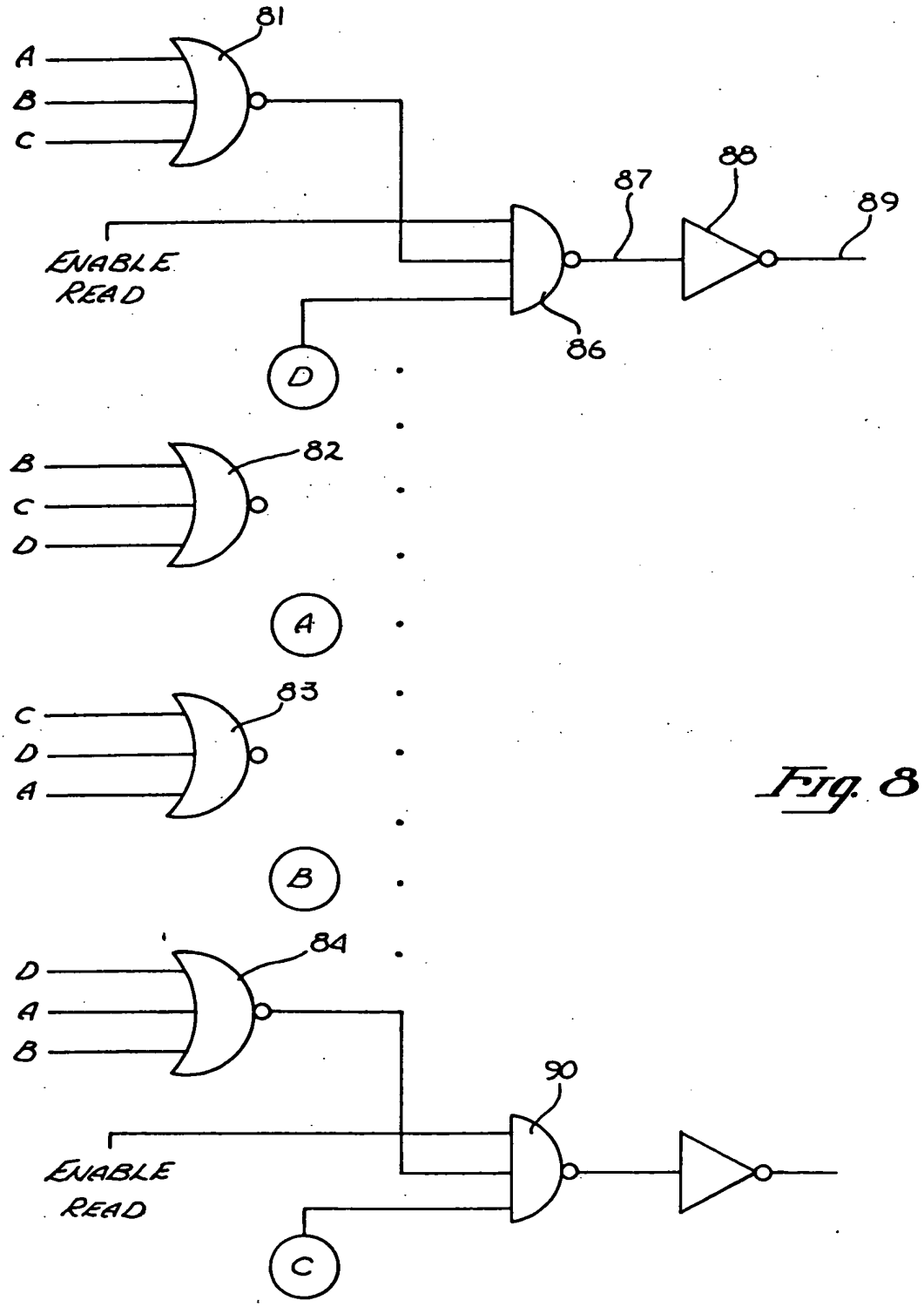
81
A
B
C
87
88
89
ENABLE
READ
86
D
82
B
C
D
A
83
C
D
A
B
84
D
A
B
90
ENABLE
READ
C

Fig. 8

SPECIFICATION

Memory management for microprocessor system

Background of the invention

1. Field of the invention.

The invention relates to the field of address translation units for memory management, particularly in a microprocessor system.

2. Prior Art

There are many well-known mechanisms for memory management. In some systems, a larger address (virtual address) is translated to a smaller physical address. In others, a smaller address is used to access a larger memory space, for instance, by using bank switching. The present invention relates to the former category, that is, where larger virtual address is used to access a limited physical memory.

In memory management systems, it is also known to provide various protection mechanisms. For example, a system may prevent a user from writing into an operating system or perhaps even from reading the operating system to external ports. As will be seen, the present invention implements a protection mechanism as part of a broader control scheme which assigns "attributes" to data on two distinct levels.

The closest prior art known to Applicant is that described in U.S. Patent 4,442,484. This patent describes the memory management and protection mechanism embodied in a commercially available microprocessor, the Intel 286. This microprocessor includes segmentation descriptor registers containing segment base addresses, limit information and attributes (e.g., protection bits). The segment descriptor table and the segment descriptor registers both contain bits defining various control mechanisms such as privilege level, types of protection, etc. These control mechanisms are described in detail in U.S. Patent 4,442,484.

One problem with the Intel 286 is that the segment offset is limited to 64k bytes. It also requires consecutive locations in physical memory for a segment which is not always easy to maintain. As will be seen, one advantage to the invented system is that the segment offset is as large as the physical address space. Yet, the invented system still provides compatibility with the prior segmentation mechanism found in the Intel 286. Other advantages and distinctions between the prior art system discussed in the above-mentioned patent and its commercial realization (Intel 286 microprocessor) will be apparent from the detailed description of the present invention.

Summary of the invention

An improvement to a microprocessor system which includes a microprocessor and a data memory is described. The microprocessor includes a segmentation mechanism for translating a virtual memory address to a second memory address (linear address) and for testing and controlling attributes of data memory segments. The improvement of the present invention includes a page cache memory on the microprocessor for translating a first field from the linear address for a hit or match condition. The data memory also stores page mapping data, specifically, a page directory and a page table. The first field accesses the page directory and page table if no hit occurs in the page cache memory. The output from either the page cache memory or the page table provide a physical base address for a page in memory. Another field of the linear address provides an offset within the page.

Both the page cache memory and page mapping data in the data memory store signals representing attributes of the data in a particular page. These attributes include read and write protection, indicate whether the page has been previously written into, and other information. Importantly, the page level protection provides a second tier of control over data in the memory which is separate and distinguished from the segment attributes.

Brief description of the drawings

Figure 1 is a block diagram showing the overall architecture of the microprocessor in which the present invention is currently realized.

Figure 2 is a block diagram illustrating the segmentation mechanism embodied in the microprocessor of Figure 1.

Figure 3 is a block diagram illustrating the page field mapping for a hit or match in the page cache memory.

Figure 4 is a block diagram illustrating the page field mapping for no hit or match in the page cache memory of Figure 3. For this condition, the page directory and page table in main memory are used and, hence, are shown in Figure 4.

Figure 5 is a diagram used to illustrate the attributes stored in the page directory, page table page cache memory.

Figure 6 is a block diagram illustrating the organization of the content addressable memory and data storage contained within the page cache memory.

Figure 7 is an electrical schematic of a portion of the content addressable memory of Figure 6.

Figure 8 is an electrical schematic of the logic circuits associated with the detector of Figure 6.

Detailed description of the present invention

A microprocessor system and in particular, a memory management mechanism for the system is described. In the following description, numerous specific details are set forth such as specific number of bits, etc., in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures are not shown in detail in order not to unnecessarily obscure the present invention.

In its currently preferred embodiment, the microprocessor system includes the microprocessor 10 of Figure 1. This microprocessor is fabricated on a single silicon substrate using complementary metal-oxide-semiconductor (CMOS) processing. Any one of many well-known CMOS processes may be employed, moreover, it will be obvious that the present invention may be realized with other technologies, for inst-

ance, n-channel, bipolar, SOS, etc.

The memory management mechanism for s me conditions requires access to tables stored in main memory. A random-access memory (RAM) 13 which functions as the main memory for the system is shown in Figure 1. An ordinary RAM may be used such as one employing dynamic memories.

As shown in Figure 1, the microprocessor 10 has a physical address of 32 bits, and the processor itself is a 32-bit processor. Other components of a microprocessor system commonly used such as drivers, mathematical processors, etc., are not shown in Figure 1.

Highlight of invention

The invented memory makes use of both segmentation and paging. Segments are defined by a set of segment descriptor tables that are separate from the page tables used to describe the page translation. The two mechanisms are completely separate and independent. A virtual address is translated to a physical address in two distinct steps, using two distinct mapping mechanisms. A segmentation technique is used for the first translation step, and a paging technique is used for the second translation step. The paging translation can be turned off to produce a one-step translation with segmentation only, which is compatible with the 286.

Segmentation (the first translation) translates a 48-bit virtual address to a 32-bit linear (intermediate) address. The 48-bit virtual address is composed of a 16-bit segment selector, and a 32-bit offset within this segment. The 16-bit segment selector identifies the segment, and is used to access an entry from the segment descriptor table. This segment descriptor entry contains a base address of the segment, the size (limit) of the segment, and various attributes of the segment. The translation steps adds the segment base to the 32-bit offset in the virtual address to obtain a 32-bit linear address. At the same time, the 32-bit offset in the virtual address is compared against the segment limit, and the type of the access is checked against the segment attributes. A fault is generated and the addressing process is aborted, if the 32-bit offset is outside the segment limit, or if the type of the access is not allowed by the segment attributes.

Paging (the second translation) translates a 32-bit linear address to a 32-bit physical address using a two-level paging table, in a process described in detail below.

The two steps are totally independent. This permits a (large) segment to be composed of several pages, or a page to be composed of several (small) segments.

A segment can start on any boundary, and be of arbitrary size, and is not limited to starting on a page boundary, or to have a length that is an exact multiple of pages. This allows segments to describe separately protected areas of memory that start at arbitrary addresses and to be of arbitrary size.

Segmentation can be used to cluster a number of small segments, each with its unique protection attributes and size, into a single page. In this case, segmentation provides the protection attributes, and paging provides a convenient method of physical memory mapping a group of related units that must be protected separately.

Paging can be used to break up very large segments into small units for physical memory management. This provides a single identifier (the segment selector), and a single descriptor (the segment descriptor) for a separately protected unit of memory, rather than requiring the use of a multitude of page descriptors. Within a segment, paging provides an additional level of mapping that allows large segments to be mapped into separate pages that need not be contiguous in physical memory. In fact, paging allows a large segment to be mapped so that only a few pages at a time are resident in physical memory, with the remaining parts of the segment mapped onto disk. Paging also supports the definition of substructure within a large segment, for example, to write protect some pages of a large segment, while other pages can be written into.

Segmentation provides a very comprehensive protection model which works on the "natural" units used by a programmer: arbitrary sized pieces of linearly addressed memory. Paging provides the most convenient method for managing physical memory, both system main memory and backing disk memory. The combination of the two methods in the present invention provides a very flexible and powerful memory protection model.

Overall microprocessor architecture

In Figure 1, the microprocessor includes a bus interface unit 14. The bus unit includes buffers for permitting transmission of the 32-bit address signals and for receiving and sending the 32 bits of data. Internal to the microprocessor, unit 14 communicates over the internal bus 19. The bus unit includes a pre-fetch unit for fetching instructions from the RAM 12 and a prefetch queue which communicates with the instruction unit of the instruction decode unit 16. The queued instructions are processed within the execution unit 18 (arithmetic logic unit) which includes a 32-bit register file. This unit, as well as the decode unit, communicate with the internal bus 19.

The present invention centers around the address translation unit 20. This unit provides two functions; one associated with the segment descriptor registers, and the other with the page descriptor cache memory. The segment registers are for the most part known in the prior art; even so, they are described in more detail in conjunction with Figure 2. The page cache memory and its interaction with the page directory and page table stored within the main memory 13 is discussed in conjunction with Figures 3-7 and forms the basis for the present invention.

Segmentation mechanism

The segmentation unit of Figure 1 receives a virtual address from the execution unit 18 and accesses the appropriate register segmentation information. The register contains the segment base address which along with the offset from the virtual address are coupled over lines 23 to the page unit.

Figure 2 illustrates the accessing of the tables in main memory when the segmentation registers are

loaded with mapping information for a new segment. The segment field indexes the segment descriptor table in the main memory 13. The contents of the table provide a base address and additionally, provide attributes associated with the data in the segment. The base address and offset are compared to the segment limits in comparator 27; the output of this comparator providing a fault signal. The adder 26 which is part of the microprocessor combines the base and offset to provide a "physical" address on lines 31. This address may be used by the microprocessor as a physical address or used by the paging unit. This is done to provide compatibility with certain programs written for a prior microprocessor (Intel 286). For the Intel 286, the physical address space is 24 bits.

The segment attributes including details on the descriptors employed such as the various privilege levels are set forth in U.S. Patent 4,442,484.

The fact that the segmentation mechanism is known in the prior art is represented in Figure 2 by the dotted line 28 which indicates the prior art structures to the left of the dotted line.

The page field mapping block 30 which includes the page unit of Figure 1 as well as its interaction with the page directory and page table stored in the main memory is shown in Figures 3 through 7.

While in the currently preferred embodiment the segmentation mechanism uses shadow registers, it also could be implemented with a cache memory as is done with the paging mechanism.

Page descriptor cache memory

In Figure 3 the page descriptor cache memory of the page unit 22 of Figure 1 is shown within dotted line 22a. This memory comprises two arrays, a content addressable memory (CAM) 34 and a page data (base) memory 35. Both memories are implemented with static memory cells. The organization of memories 34 and 35 is described in conjunction with Figure 6. The specific circuitry used for CAM 34 with its unique masking feature is described in conjunction with Figures 7 and 8.

The linear address from the segment unit 21 are coupled to the page unit 22 of Figure 1. As shown in Figure 3, this linear address comprises two fields, the page information field (20 bits) and a displacement field (12 bits). Additionally, there is a four bit page attribute field provided by the microcode. The 20-bit page information field is compared with the contents of the CAM 34. Also, the four attribute bits ("dirty", "valid", "U/S", and "W/R") must also match those in the CAM before a hit occurs. (There is an exception to this when "masking" is used as will be discussed.)

For a hit condition, the memory 35 provides a 20-bit base word which is combined with the 12-bit displacement field of the linear address as represented by summer 36 of Figure 3 and the resultant physical address selects from a 4k byte page frame in main memory 13.

Page addressing for the no-hit condition

A page directory 13a and a page table 13b are stored in the main memory 13 (see Figure 4). The base address for the page directory is provided from the microprocessor and is shown in Figure 4 as the page directory base 38. Ten bits of the page information field are used as an index (after being scaled by a fact r of 4) into the page directory as indicated by the summer 40 in Figure 4. The page directory provides a 32-bit word. Twenty bits of this word are used as a base for the page table. The other 10 bits of the page information field are similarly used as an index (again being scaled by a factor of 4) into the page table as indicated by the summer 41. The page table also provides a 32-bit word, 20 bits of which are the page base of the physical address. This page base address is combined as indicated by summer 42 with the 12-bit displacement field to provide a 32-bit physical address.

Five bits from the 12-bit fields of the page directory and table are used for attributes particularly "dirty", "access", "U/S, "R/W" and "present". These will be discussed in more detail in conjunction with Figure 5. Remaining bits of this field are unassigned.

The stored attributes from the page directory and table are coupled to control logic circuit 75 along with the 4 bits of attribute information associated with the linear address. Parts of this logic circuit are shown in subsequent figures are discussed in conjunction with these figures.

Page directory attributes

In Figure 5 the page directory word, page tabl word and CAM word are again shown. The protectiv / control attributes assigned to the four bits of the pag directory word are listed within bracket 43. The same four attributes with one additional attribute are used for the page table word and are set forth within bracket 44. The four attributes used for the CAM word are set forth within bracket 45.

The attributes are used for the following purpose:

1. DIRTY. This bit indicates whether a page has been written into. The bit is changed once a pag has been written into. This bit is used, for instance, t inform the operating system that an entire page is not "clean". This bit is stored in the page table and in the CAM (not in the page directory). The processor sets this bit in the page table when a page is written into.
2. ACCESSED. This bit is stored in only the page directory and table (not in the CAM) and is used to indicate that a page has been accessed. Once a page is accessed, this bit is changed in the memory by the processor. Unlike the dirty bit, this bit indicates whether a page has been accessed either for writing or reading.
3. U/S. The state of this bit indicates whether the contents of the page is user and supervisory accessible (binary 1) or supervisor only (binary zero).
4. R/W. This read/write protection bit must be a binary 1 to allow the page to be written into by a user level program.
5. PRESENT. This bit in the page table indicates if the associated page is present in the physical memory. This bit in the page directory indicates if the associated page table is present in physical memory.
6. VALID. This bit which is stored only in the CAM is used to indicate if the contents of the CAM is valid. This bit is set to a first state on initialization then changed when a valid CAM word is loaded.

The five bits from the page directory and table are coupled to control logic circuit 75 to provide appropriate fault signals within the microprocessor.

The user/supervisor bits from the page directory and table are logically ANDed as indicated by gate 46 to provide the R/W bit stored in the CAM 34 of Figure 3. Similarly, the read/write bits from the page directory and table are logically ANDed through gate 47 to provide the W/R bit stored in the CAM. The dirty bit from the page table is stored in the CAM. These gates are part of the control logic 75 of Figure 4.

The attributes stored in the CAM are "automatically" tested since they are treated as part of the address and matched against the four bits from the microcode. A fault condition results even if a valid page base is stored in the CAM, if, for instance, the linear address indicates that a "user" write cycle is to occur into a page with R/W=0.

The ANDing of the U/S bits from the page directory and table ensures that the "worst case" is stored in the cache memory. Similarly, the ANDing of the R/W bit provides the worst case for the cache memory.

Organization of the page descriptor cache memory

The CAM 34 as shown in Figure 6 is organized in 8 sets with 4 words in each set. Twenty-one bits (17 address and 4 attributes) are used to find a match in this array. The four comparator lines from the four stored words in each set are connected to a detector. For instance, the comparator lines for the four words of set 1 are connected to detector 53. Similarly, the comparator lines for the four words in sets 2 through 8 are connected to detectors. The comparator lines are sensed by the detectors to determine which word in the set matches the input (21 bits) to the CAM array. Each of the detectors contains "hard wired" logic which permits selection of one of the detectors depending upon the state of the 3 bits from the 20-bit page information field coupled to the detectors. (Note the other 17 bits of this bit page information field is coupled to the CAM array.)

For purposes of explanation, eight detectors are implied from Figure 6. In the current embodiment only one detector is used with the three bits selecting one set of four lines for coupling to the detector. The detector itself is shown in Figure 8.

The data storage portion of the cache memory is organized into four arrays shown as arrays 35a-d. The data words corresponding to each set of the CAM are distributed with one word being stored in each of the four arrays. For instance, the data word (base address) selected by a hit with word 1 of set 1 is in array 35a, the data word selected by a hit with word 2 of set 1 is in array 35b, etc. The three bits used to select a detector are also used to select a word in each of the arrays. Thus, simultaneously, words are selected from each of the four arrays. The final selection of a word from the arrays is done through the multiplexer 55. This multiplex r is controlled by the four comparator lines in the detector.

When the memory cache is accessed, the matching process which is a relatively slow process begins through use of the 21 bits. The other three bits are able to immediately select a set of four lines and the detector is prepared for sensing a drop in potential on the comparator lines. (As will be discussed, all the comparator (rows) lines are precharged with the selected (hit) line remaining charged while the non-selected lines discharge.) Simultaneously, four words from the selected set are accessed in arrays 35a-35d. If and when a match occurs, the detector is able to identify the word within the set and this information is transmitted to the multiplexer 55 allowing the selection of the data word. This organization improves access time in the cache memory.

Content addressable memory (CAM)

In Figure 7, the 21 bits which are coupled to the CAM array are again shown with 17 of the bits being coupled to the complement generator and override circuit 56 and with the 4 attribute bits coupled to the VUDW logic circuit 57. The 3 bits associated with the selection of the detectors described in conjunction with Figure 6 are not shown in Figure 7.

The circuit 56 generates the true and complement signal for each of the address signals and couples them to parallel lines in the CAM array, such as lines 59 and 60. Similarly, the VUDW logic 57 generates both the true and complement signals for the attribute bits and couples them to parallel lines in the array. The lines 59 and 60 are duplicated for each of the true and complement bit lines (i.e., 21 pairs of bit and bit/ lines).

Each of the 32 rows in the CAM array has a pair of parallel row lines such as lines 68 and 70. An ordinary static memory cell such as cell 67 is coupled between each of the bit and bit/ lines (columns) and is associated with the pair of row lines. In the presently pref r-red embodiment, the memory cells comprise ordinary flip-flop static cells using p-channel transistors. One line of each pair of row lines (line 70) permits the memory cell to be coupled to the bit and bit/ line when data is written into the array. Otherwise, the content of the memory cell is compared to the data on the column lines and the results of the comparison is coupled to the hit line 68. The comparison is done by comparators, one associated with each cell. The comparator comprises the n-channel transistors 61-64. Each pair of the comparator transistors, for example, transistors 61 and 62, are coupled between one sid of the memory cell and the opposite bit line.

Assume that data is stored in the memory cell 67 and that the node of the cell closest to bit line 59 is high. When the contents of the CAM are examined, first the hit line 68 is precharged through transistor 69. Then the signals coupled to the CAM are placed on the column lines. Assume first that line 59 is high. Transistor 62 does not conduct since line 60 is low. Transistor 63 does not conduct since the side of the cell to which it is connected is low. For these conditions, line 68 is not discharged, indicating that a match has occurred in the cell. The hit line 68 provides ANDing of the comparisons occurring along the row. If a match does not occur, one or more of the comparators will cause the hit line to discharge.

During precharging the circuits 56 and 57 generate an override signal causing all column lines (both bit and bit/) to be low. This prevents the comparators from draining the charge from the hit lines before the comparison begins.

It should be noted that the comparators examine the "binary one" condition and, in effect, ignore the "binary zero" condition. That is, for instance, if the gate of transistor 64 is high (line 59 high) then transistors 63 and 64 control the comparison. Similarly, if the bit/line 60 is high, then transistors 61 and 62 control the comparison. This feature of the comparator permits cells to be ignored. Thus, when a word is coupled to the CAM, certain bits can be masked from the matching process by making both the bit and bit/line low. This makes it appear that the contents of the cell match the condition on the column lines. This feature is used by the VUDW logic circuit 57.

Microcode signals coupled to logic circuit 57 causes the bit and bit/line for selected ones of the attribute bits to be low as a function of the microcode bits. This results in the attribute associated with that bit to be ignored. This feature is used, for instance, to ignore the U/S bit in the supervisory mode. That is, the supervisory mode can access user data. Similarly, the read/write bit can be ignored when reading or when the supervisory mode is active. The dirty bit is also ignored when reading. (The feature is not used for the valid bit.)

When the attribute bits are stored in main memory, they can be accessed and examined and logic circuits used to control accessing, for instance, based on the one or zero state of the U/S bit. However, with the cache memory no separate logic is used. The forcing of both the bit and bit/lines low, in effect, provides the extra logic by allowing a match (or preventing a fault) even though the bit patterns of the attribute bits are not matched.

The detector from Figure 6, as shown in Figure 8, includes a plurality of NOR gates such as gates 81, 82, 83 and 84. Three of the hit lines from the selected set of CAM lines are coupled to gate 81; these are shown as lines A, B, and C. A different combination of the lines are connected to each of the other NOR gates. For instance, NOR gate 84 receives the hit lines D, A, and B. The output of each of the NOR gates is an input to a NAND gate such as NAND gate 86. A hit line provides one input to each NAND gate. This line is the one (of the four A,B,C,D) that is not an input to the NOR gate. This is also the bit line from the set entry to be selected. For example, gate 86 should select the set that is associated with hit line D. For instance, in the case of NOR gate 81, hit line D is coupled to the NAND gate 86. Similarly, for the NAND gate 90, the hit line C in addition to the output of gate 84, are inputs to this gate. An enable read signal is also coupled to the NAND gates to prevent the outputs of this logic from being enabled for a write. The output of the NAND gates, such as line 87, are used to control the multiplexer 55 of Figure 6. In practice, the signal from the NAND gate, such as the signal on line 87, controls the multiplexer through p-channel transistors. For purposes of explanation, an additional inverter 88 is shown with an output line 89.

The advantage to this detector is that it enables precharge lines to be used in the multiplexer 55. Alternately, a static arrangement could be used, but this would require considerably more power. With th arrangement as shown in Figure 8, th output from the inverters will remain in the same state until one of the hit lines drops in potential. When that occurs, only a single output line will drop in potential, permitting the multiplexer to select the correct word.

Thus, a uniqu address translation unit has been described which uses two levels of cache memory, one for segmentation and one for paging. Independent data attribute control (e.g., protection) is provided on each level.

CLAIMS

1. In a microprocessor system which includes a microprocessor and a data memory where the microprocessor has a segmentation mechanism for translating a virtual memory address to a second memory address and for controlling data based on attributes, an improvement comprising:

a page cache memory integral with said microprocessor for receiving a first field of said second memory address and for comparing it with contents of said page cache memory to provide a second field under certain conditions;

said data memory including storage for page mapping data, said first field of said second memory address being coupled to said data memory to sel ct a third field from said page data when said certain conditions of said page cache memory are not met;

said microprocessor system including a circuit for combining one of said second and third fields with an offset field from said first address to provide a physical address for said data memory;

whereby the physical addressibility of said data memory is improved.

2. The improvement defined by Claim 1 wherein said page cache memory and said storage for said page data includes information on the attributes of memory pages.

3. The improvement defined by Claim 2 wherein said storage for said page mapping data comprises at least one page directory and at least one page tabl .

4. The improvement defined by Claim 3 wherein each of said page directory and said page table store said attributes for said memory pages.

5. The improvement defined by Claim 4 wherein at least some of said attributes stored in said page directory and said page table are logically combined and stored in said page cache memory.

6. The improvement defined by Claim 5 wherein said microprocessor provides a page directory base for said page directory.

7. The improvement defined by Claim 6 wherein a first portion of said first field provides an index into said page directory base to a location in said page directory.

8. The improvement defined by Claim 7 wherein said locations in said page directory store page table bases and wherein a second portion of said first field provides an index into said page table to a page table location in said data memory.

9. The improvement defined by Claim 8 wherein said locations in said page table provide a base to pages in said data memory.

10. The improvement defined by Claim 2 wherein said page cache memory includes a content addressable memory (CAM) and a page base memory, the

output of said CAM selecting page bases for said data memory from said page base memory.

11. The improvement defined by Claim 10 wherein said CAM stores attributes of data memory pages.

12. The improvement defined by Claim 11 wherein said CAM includes means for selectively masking at least one of said attributes during said comparison.

13. An improvement in memory management for a microprocessor system comprising:

a microprocessor having a segmentation mechanism for translating a virtual memory address to a second memory address and for testing attributes of data memory segments;

a data memory coupled to said microprocessor;

said microprocessor including a page cache memory integral with said microprocessor for receiving a first field of said second memory address and for comparing it with contents of said cache memory to provide a second field under certain conditions;

said data memory including storage for page mapping data, said first field of said second memory address being coupled to said data memory to select a third field from said page data when said certain conditions of said page cache memory are not met;

said microprocessor system including a circuit for combining one of said second and third fields with an offset field from said first address to provide a physical address for said data memory;

whereby the physical addressibility of said data memory is improved.

14. The improvement defined by Claim 13 wherein said segmentation mechanism comprises:

segment descriptor registers integral with said microprocessor for providing a segment base; and

said data memory including a segment descriptor table which is accessed by a segment field of said first address.

15. The improvement defined by Claim 14 wherein said page cache memory and said storage for said page data includes information on the attributes of memory pages.

16. The improvement defined by Claim 15 wherein said storage for said page mapping data comprises a page directory and a page table.

17. The improvement defined by Claim 16 wherein each of said page directory and page table store said attributes for said memory pages.

18. The improvement defined by Claim 17 wherein at least some of said attributes stored in said page directory and page table are logically combined and stored in said page cache memory.

19. An address translation unit formed as part of a microprocessor for operating with a data memory comprising:

segment descriptor registers for receiving a virtual address and for providing a segment base;

said microprocessor for providing an address for the data memory to permit addressing of a segment descriptor table in said data memory, said segment descriptor table providing said segment base address;

said microprocessor employing said second base address and a portion of said virtual address to provide a second memory address;

a page cache memory for receiving a first field of said second memory address and for comparing it with the contents of said page cache memory to provide a second field under certain second conditions;

said microprocessor for providing said first field to a page data table in said data memory for providing said second field if said second conditions are not met;

said second field providing a page base for said data memory,

whereby the physical addressibility of said data memory is improved.

20. The unit defined by Claim 19 wherein said segment descriptor registers store segment data attributes and wherein said page cache memory stores page data attributes.

21. A content addressable memory (CAM) comprising:

a plurality of buffers, each for receiving first signals and for providing said first signals and second signals, said second signals being complements of said first signals;

a plurality of a generally parallel pairs of lines each pair being coupled to receive one of said first and second signals;

a plurality of memory cells coupled between each pair of lines said cells being arranged in rows generally perpendicular to said pairs of lines;

a plurality of row comparator lines one associated with each of said rows of cells;

a plurality of comparators, one for coupling between each of said memory cells, its respective pair of lines and one of said comparator lines, said comparators for comparing a binary state stored in said memory cell with said first and second signals;

loading means for loading data from said pairs of lines to said cells;

said comparators being disabled when its respective pairs of lines are both maintained at a certain binary state;

whereby by causing at least some of said buffers to provide said certain binary state for said first and second signals, selected ones of said cells can be ignored for said comparison.

22. The CAM defined by Claim 21 wherein said row comparator lines are precharged lines.

23. The CAM defined by Claim 22 including a storage memory which comprises a plurality of secti ns and wherein data is accessed simultaneously in all of said sections and an output from one of said sections being selected through said row lines.

24. The CAM defined by Claim 23 including detectors coupled to a predetermined number of said r w lines, said detectors for sensing which one of said predetermined number of lines remains charged.

25. The CAM defined by Claim 24 wherein said selection of said output from one of said sections is made by said d tectors.

26. An improvement in memory management for a microprocessor system substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

Printed in the UK for HMSO, D8818935, 11/86, 7102.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

ERRATUM

SPECIFICATION NO. 2176918A

Page No 6 Line No 126 after drawings. Start new paragraph insert

Amendments to the claims have been filed, and have the following effect:-

Claims 21 to 25 above have been deleted.
Claim 26 above has been re-numbered as 21.

THE PATENT OFFICE
12 June 1987